

## Dokumentace: Hra Buch!

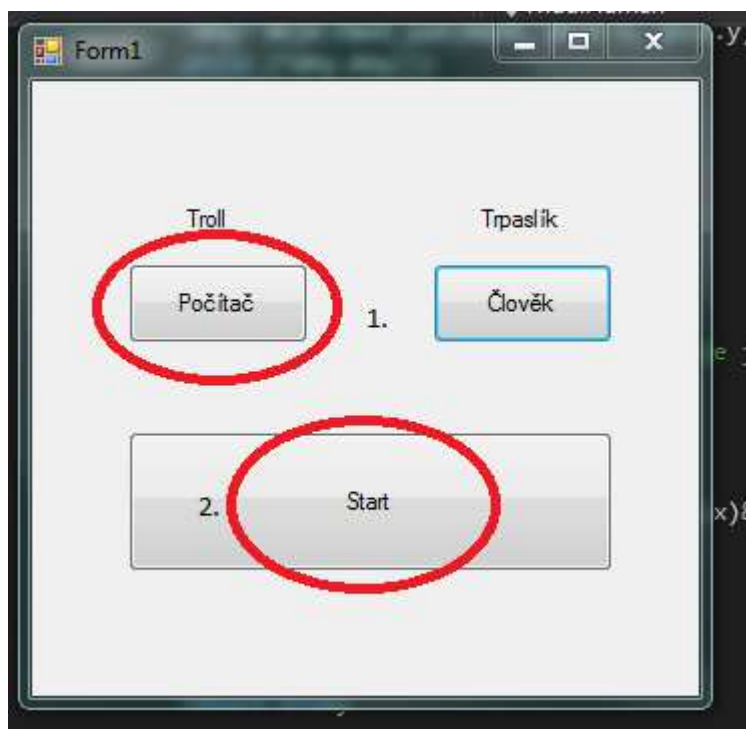
### Zadání

Implementace stolní hry "Buch!" do grafického prostředí s počítačem schopným hrát jak za Trolly, tak za Trpaslíky.

### Kompatibilita

Zaručena pro Windows 7.

### Uživatelská část



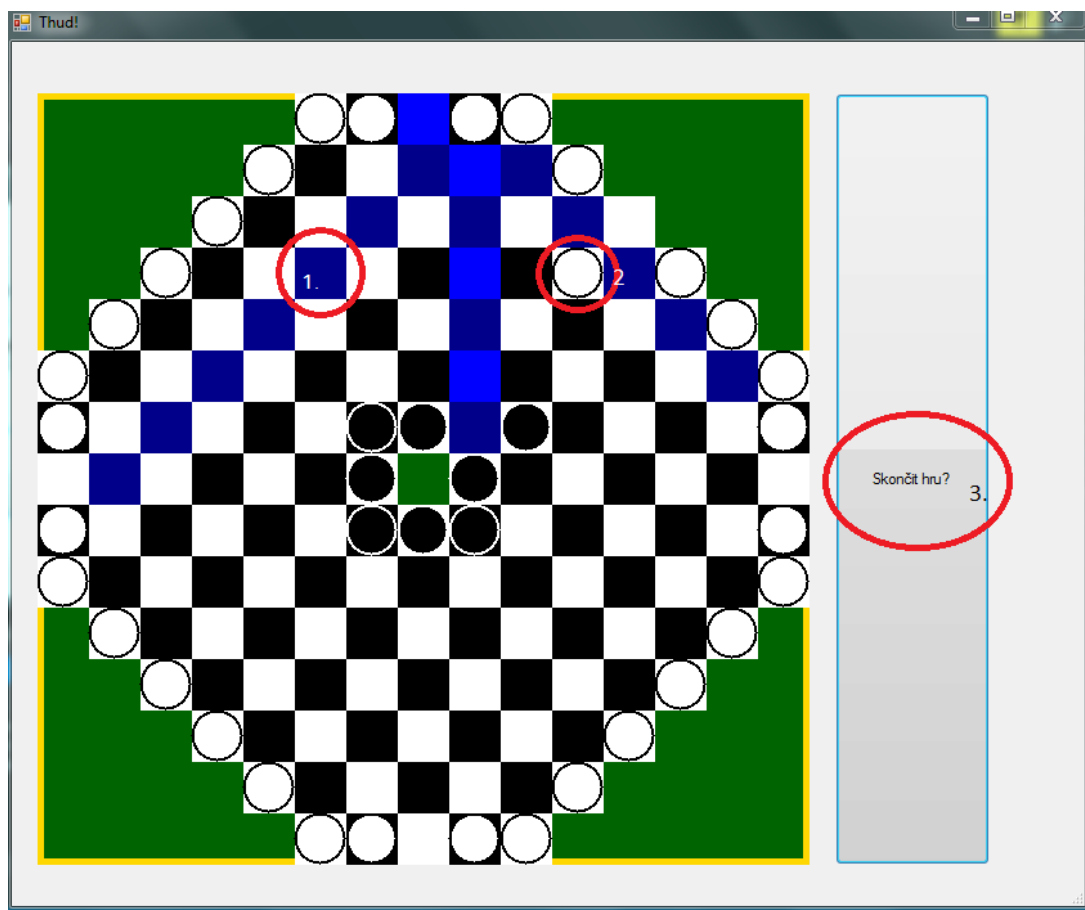
1. Kliknutím na tlačítko uživatel přepíná zda za danou strnu bude hrát počítač nebo uživatel.
2. Tlačítko Start spouští novou hru.

Hra se ovládá pouze pomocí myši.

Hráč na tahu si levým kliknutím vybírá jednu ze svých figur a levým kliknutím na vyznačené (modré) pole ji pošle na místo určení. Vyznačeny jsou právě pole, kam má dle pravidel figurka právo se pohnout.

Tím končí tah a na tahu je další hráč.

Kliknutí mimo zruší výběr figury.



1. Označené políčko možného postupu.
2. Hrací kámen (Trpaslík)
3. Tlačítko, kterým navrhuje hráč na tahu protistraně konec hry. Dialogové okno se zeptá napřed na potvrzení hráčova úmyslu hru ukončit. Případné druhé dialogové okno patří druhému hráči.

Hra si vytváří log odehraných tahů do souboru log.txt.

### ***Základní pravidla a principy:***

Hru hrají dva hráči.

Jeden hráč hraje na straně trollů, druhý na straně trpaslíků.

Černé figurky představují trolly, bílé představují trpaslíky.

V základu se troll pohybuje jako šachový král.

Trpaslíci se pohybují jako šachové dámy.

Ani trollové ani trpaslíci se nemohou pohnout přes ("přeskočit") jiný hrací kámen.

Trollové nesmí za žádných okolností vstoupit na pole, kde již leží nějaká figura.

Pokud troll vstoupí na některé pole, sebere všechny trpaslíky, kteří se nachází na jednom z

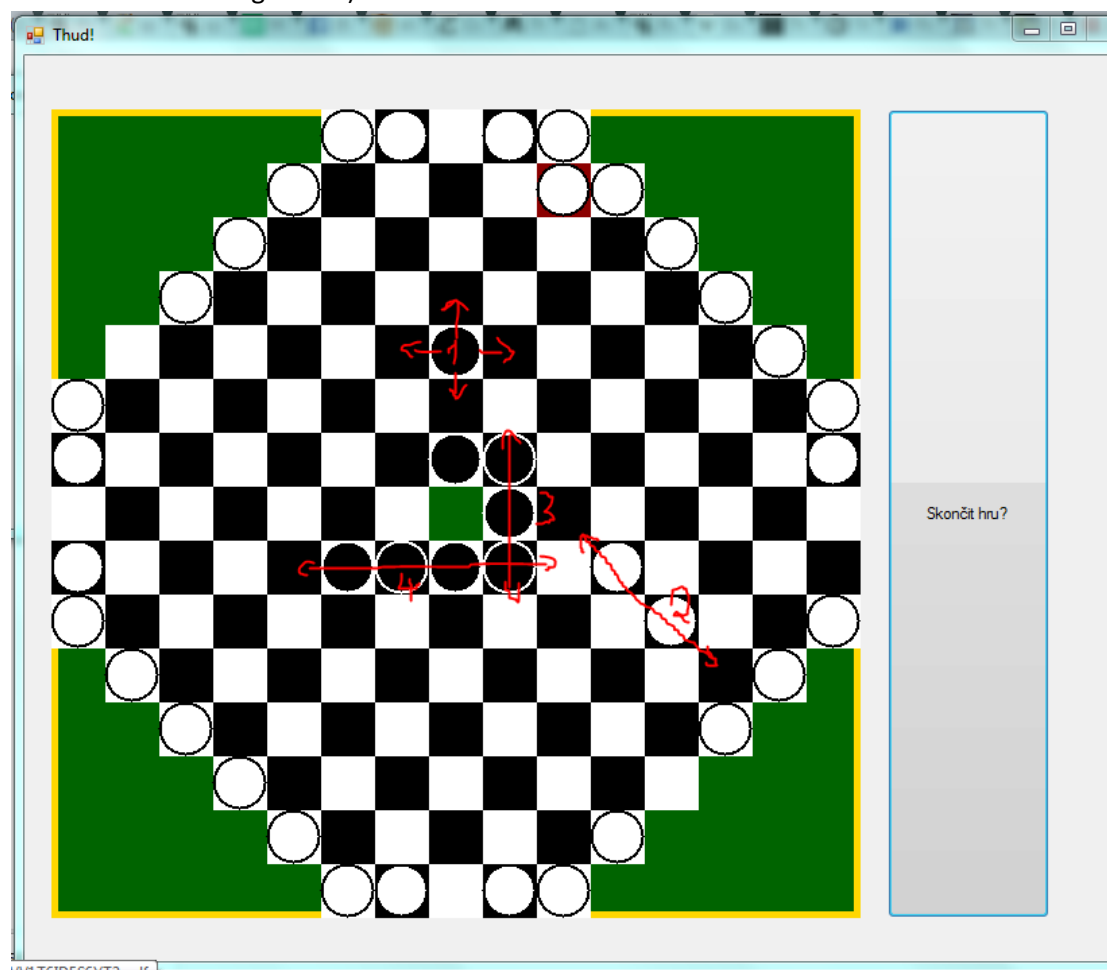
přilehlých 8 polích u místa, kam vstoupil.

*(pozn.: Vzhledem k tomu, že nevlastním originál hry a tedy ani originál pravidel, je tohle pravidlo poněkud kontroverzní. Kromě jediné výjimky, ze všech zdrojů vyplývá, že hráč všechny okolní trpaslíky sebrat **může**.*

*Některé zdroje ovšem uvádí, že sebrání závisí na volbě hráče a hráč by tedy mohl některé trpaslíky ušetřit, což ale podle mého názoru z principu hry nemá opodstatnění a hráči to může pouze uškodit.*

*Jediný zdroj (Wikipedie) pak tvrdí, že při běžném pohybu může troll sebrat pouze jednoho trpaslíka a pouze při postrku všech "okolních 8". Nikde jinde jsem však zmínku o odlišnosti systému braní figur při pohybu a při postrku nenašel. Navíc toto pravidlo je poněkud podivné díky systému počítání figurek v řadě, který by měl trollovi dovolovat, aby se postrčil sám...)*

Řadu figur tvoří figury spolu sousedící, které stojí na políčkách v jedné přímce (tedy svisle, vodorovně nebo diagonálně).



Čísla udávají kolik figur je v řadě, šipky udávají v jakých směrech je možno postrk nebo výskok uplatnit.

Pokud jsou figurky trollů v dostatečně dlouhé řadě za sebou může hráč udělat tzv. "postrk". V rámci postrku se může troll, stojící na jednom konci řady, pohnout až o tolik polí ve směru

od řady, kolik je trollů za ním (včetně jeho samého). K postrku může dojít pouze, pokud tímto tahem může troll sebrat některého trpaslíka.

Pokud jsou trpaslíci v dostatečně dlouhé řadě, pak může hráč zahrát tzv. "výskok", obdobu "postrku".

V rámci výskoku může trpaslík, stojící na konci řady sebrat trolla až do vzdálenosti tolika polí, kolik trpaslíků je v řadě za ním (včetně "vyskakujícího" trpaslíka).

Sebrání nepřátelského kamene trpaslík provede tak, že vstoupí na pole, kde leží tento nepřátelský kámen.

Pokud trpaslík neprovádí "výskok" a pouze se pohybuje, nemůže sebrat nepřátelskou figuru. Jeden osamocený trpaslík může provádět výskok o jedno pole.

Hra končí ve chvíli, kdy se hráči shodnou na tom, že nelze sebrat žádnou další figuru.

Hru vyhrává hráč s nejvyšším skóre. Skóre se počítá z přeživších figur.

Přeživší trpaslík se jeho hráči počítá za 1 bod.

Přeživší troll za 4 body.

Základní taktika trollů je rozprostřít se po hracím plánu a využít tak co nejvíce schopnost brát až 8 figur najednou.

Trpaslíci se snaží postupně uzavřít trolly do pasti, kde je budou moci sebrat. Díky vyšší mobilitě a větším počtu mohou snáze vytvářet řady a ohrožovat tak větší území. Oblíbená taktika je tzv. "falanga", kdy se trpaslíci sešikují do zhruba čvercového útvaru a mohou účeněji na trolly dorážet.

## Struktura programu

Klinutím na tlačítko Start ve Form1 je vytvořena a zobrazena instance třídy Form2.

Při vytváření instance Form2:

Jsou vytvořeny instance tříd od **Troll** a **Dwarf**, které slouží jako model pro pohyb figur po mapě.

Jsou vytvořeny dvě instance podtříd třídy **Hrac**, **Computer** nebo **Human**, v závislosti na volbě předané z hlavního menu. Těmto třídám jsou příslušným způsobem předány instance Troll a **Dwarf**.

Je vytvořena instance třídy **Deska** která se stará o zpracování grafické stránky hry.

Je vytvořena instance třídy **Rozhodci**. Konstruktoru této třídy jsou příslušným způsobem předány instance podtříd abstraktní třídy **Hrac** a třídy **Deska**.

Přes třídu Rozhodci je zahájena hra spuštěním tahu prvního hráče na vedlejším vlákně, na kterém probíhají všechny výpočty, které potřebuje hráč (především v instanci třídy

**Computer**) vykonat mimo hlavní vlákno.

**Rozhodci** reaguje na události vyvolané z instancí podtříd třídy **Hrac** a podle nich řídí průběh hry.

### **Třída Deska**

Stará se o grafickou podobu herní plochy: velikost, barvy použité pro vykreslení apod.. Obsahuje bitmapu právě hrané hry a nevlastní odkazy na žádné zobrazovací prvky. Pro komunikaci s vnějškem spouští událost *OnUpdate*. Obraz se vytváří na základě mapy tvořenou dvojrozměrným polem proměnných typu *Field*.

### **Třída Troll a Dwarf(podtřídy Figury)**

Umožňují větší univerzalitu tříd **Human** a **Computer**. V těchto třídách jsou používány pro získávání seznamu míst, kam se může figurka na zadané pozici pohnout a na vyhodnocení takového pohybu. Zvenku tyto dvě podtřídy využívá i i třída **Rozhodčí**, kde se pomocí jejich instancí vyhodnocují pohyby figur.

### **Třída Human(podtřída Hrac)**

Třída poskytuje rozhraní pro komunikaci mezi uživatelsky interaktivními prvky a hrou, tedy třídou **Rozhodci**.

### **Třída Rozhodci**

Úkolem této třídy je řídit průběh hry.

Funkce pak zpracuje předávání tahu druhému hráči a aktualizaci mapy. Hráčské funkce zpracovávající tah spouští ve zvláštním vláknu.

Objekt této třídy také zaznamenává průběh hry a stará o vytvoření a rozložení hracího plánu, na který poskytuje hráčským instancím odkaz.

### **Třída Computer**

Třída implementuje objekt umělého protivníka.

Algoritmus funguje na principu pohledávání do hloubky.

Pro vyhodnocení jednotlivých větví je použitý algoritmus Negamax s alfa-beta ořezáváním.

Vzhledem k značnému faktoru větvení hry, v průměru přes 300, v maximu i kolem 650, je toto prohledávání poměrně náročné a v lidsky únosném čase (1-2minuty) dokáže bot vidět maximálně pouze do hloubky 3 tj. 2 celé tahy obou hráčů dopředu.

Pro hloubku 4 je zatím nevyzkoušený odhad 3-4 hodiny/tah. pro srovnání průměrný faktor větvení pro šachy je dle wikipedie 35 - 38, pro hru Go pak kolem 250. Počítání skóre koncových uzlů probíhá na základě zdrojů, tedy  $score = počet\_mých\_figur - počet\_nepřátelských\_figur$ .

## **Alternativy algoritmů**

Vzhledem k malé hloubce prohledávání a jistému skórovacímu znevýhodnění trpaslíků, kteří v rámci 2 tahového výhledu nemají téměř šanci na nalezení tahu s kladným skóre, byli některé zrychlovací taktiky jako nulové okno z programování šachových programů velice neúčinné a téměř neměli vliv na výkon (a pokud měli pak spíše negativní).

Dobrou alternativou by nejspíš bylo počítání skóre trpaslíků spíše na základě pozice než pouze na základě zdrojů, čímž by se mohla kompenzovat nevýhoda malého rozhledu. Problém to ale zcela neřeší.

Nabízí to ovšem také otázku co je vlastně v Buch! dobrá pozice a jak to zjistit.

V rámci výběru tahů by možná k zmenšení objemu zpracovávaných tahů využít poměrně velké symetrie hracího plánu.

Jako nejúčinnější alternativa se ale jeví nějaký učící se algoritmus.

## **Průběh prací**

1. Objektový návrh programu.
  2. Vytýčení kostry tříd Hrac, Figura, Rozhodci
  3. Implementace tříd Troll a Dwarf
  4. Implementace třídy Deska
  5. Implementace třídy Human a Rozhodci
  6. Propojení tříd s ovládacím prvky na Form2
  7. Testování a úpravy tříd Human, Rozhodčí a Deska
  8. Implementace třídy Computer
  9. Testování třídy Computer a úpravy tříd Rozhodci a Human pro potřeby běhu hráčských výpočtů na dalším vlákně.
- Implementace hlavního menu v podobě Form1
10. Testy a závěrečné úpravy třídy Computer

## **Návrhy na zlepšení**

Vylepšení, které rozhodně plánuji někdy dodělat je přidat hodnocení "zajímavosti" tahu, které by při rozhodování mezi stejně obodovanými tahy upřednostnilo ten "nejzajímavější" tj. takový který buď vytváří řadu, nebo se jím přilíže figura k nepříteli apod. Takovéto tahy možná nemusí zvýšit předpoklad na výhru, ale alespoň to může předejít kixům, kdy se při hře 2 počítačů oba posouvají o minimální vzdálenost a "a krčí se v koutě".

Jak již bylo zmíněno výše, trpaslíci jsou teď poněkud znevýhodněni, jelikož jejich hra závisí především na dlouhodobém plánu a schopnosti obětovat málo pro zisk v budoucnu. To se jak duřám dá zlepšit, buď zvýšením výkonu algoritmu, např implementací "naučených"

reakcí na určité konstelace herního plánu, a tedy prodloužením výhledového okna, nebo výše zmíněnou úpravou skórovací funkce.

## **Závěr**

Ve chvíli kdy jsem pouštěl do projektu, čekal jsem že se bude jednat o výpočetně náročnější hru než šachy, rozhodně nečekal, že se z Buch! vyklube tak komplexní hra. Možná jsem ale rád za časovou náročnost jejich výpočtů, jelikož jsem kromě náhledu na to jak to v C# funguje s událostmi musel prozkoumat práci s více vlákny, která se navzájem zkze volání událostí mají prolínat.

Díky práci na projektu jsem si uvědomil některé skutečnosti o programování rozhodovacích algoritmů a dozvěděl se víc o metodách, které se při něm dají použít.

V budoucnu bych se rád zabýval rozhodovacími algoritmy podrobněji.

A pokud už nic jiného, alespoň už vím na jisto proč se všechny implementace této hry dají hrát pouze s lidským protivníkem.